

# LEHRPLAN

---

# INFORMATIK

Gymnasialer Bildungsgang

Gymnasiale Oberstufe

HESSEN



Hessisches Kultusministerium

2010

Inhaltsverzeichnis		Seite
<b>Teil A</b>	<b>Grundlegung für das Unterrichtsfach Informatik in der Einführungsphase und der Qualifikationsphase</b>	3
1	Aufgaben und Ziele des Faches	3
2	Didaktisch-methodische Grundlagen	5
2.1	Leitlinien	5
2.2	Arbeitsweisen	6
2.3	Grund- und Leistungskurse	6
3	Umgang mit dem Lehrplan	7
3.1	Verbindliche Vorgaben	7
3.2	Lern- und Prüfungsbereiche	8
<b>Teil B</b>	<b>Unterrichtspraktischer Teil Der Unterricht in der Sekundarstufe II</b>	9
	Übersicht der verbindlichen Themen	9
1	Die verbindlichen und fakultativen Unterrichtsinhalte der Einführungs- und der Qualifikationsphase	10
1.1	Die Einführungsphase (E1 und E2)	10
1.1.1	E1	11
1.1.2	E2	14
1.2	Die Qualifikationsphase (Q1 bis Q4)	17
1.2.1	Q1	17
1.2.2	Q2	20
1.2.3	Q3	22
1.2.4	Q4	26
2	Abschlussprofil am Ende der Qualifikationsphase	34

## **Teil A      Grundlegung für das Unterrichtsfach Informatik in der Einführungsphase und der Qualifikationsphase**

### **1      Aufgaben und Ziele des Faches**

Der umfassende Einsatz der Informations- und Kommunikationstechniken hat in Wirtschaft, Verwaltung, Technik und Wissenschaft zu tiefgreifenden Veränderungen der gesellschaftlichen Strukturen geführt; alle Bereiche des Lebens sind von dieser Entwicklung betroffen. Durch die Vernetzung der Informationsquellen über das Internet ist prinzipiell die global verteilte Information für jeden Menschen zu jeder Zeit und an jedem Ort verfügbar. Zur Bewältigung der Fülle von Informationen sind Informatiksysteme unerlässlich. Mit den Diensten des Internets sind weltweite Kommunikation und Kooperation in gesellschaftlichen und individuellen Zusammenhängen möglich.

In der Informations- und Wissensgesellschaft spielen komplexe Informatiksysteme eine wachsende Rolle im täglichen Leben und verändern in zunehmendem Maße die Arbeits- und Lebensweise der Menschen. Die Informatik als Bezugswissenschaft nimmt in diesem gesellschaftlichen Wandel eine Schlüsselrolle ein, da sie sich mit den theoretischen Grundlagen, Methoden und Verfahren der Informations- und Kommunikationstechniken beschäftigt, welche auch gegenüber der ungemein schnellen technischen Entwicklung Bestand haben. Der verantwortungsbewusste Umgang mit Information, das Verständnis für dynamische Abläufe und für entsprechende Problemlösungsmethoden setzen grundlegende Kenntnisse der Informatik für alle voraus. Der kompetente Umgang mit digitaler Information und mit Informatiksystemen stellt eine unverzichtbare Ergänzung der traditionellen Kulturtechniken Lesen, Schreiben und Rechnen dar und ist wichtiger Bestandteil einer allgemeinen Studier- und Berufsfähigkeit. Es ist Aufgabe der allgemein bildenden Schule, allen Schülerinnen und Schülern unabhängig von ihrem Geschlecht, ihrer Herkunft und ihren sozialen Verhältnissen einen gleichberechtigten Zugang zu informatischen Denk- und Arbeitsweisen und modernen Informations- und Kommunikationstechniken zu öffnen, informatische Bildung zu vermitteln und damit auch eine Grundlage für lebenslanges Lernen zu schaffen.

Das Schulfach Informatik ergänzt die für eine zeitgemäße Allgemeinbildung erforderliche Betrachtung der Welt um informatische Aspekte und leistet damit neben den naturwissenschaftlichen Fächern und der Mathematik einen spezifischen, eigenständigen Beitrag im mathematisch-naturwissenschaftlich-technischen Aufgabenfeld. Die Methoden und Werkzeuge der Informatik, die Denk- und Herangehensweisen sowie die Nutzung von Informatiksystemen findet inzwischen in fast allen Gebieten von Wissenschaft, Wirtschaft und Technik Eingang – die Geistes- und Gesellschaftswissenschaften eingeschlossen.

Der Informatikunterricht leistet einen wesentlichen Beitrag zum Bildungs- und Erziehungsauftrag der Schule, da er sich mit den Grundlagen und Anwendungen der Informations- und Kommunikationstechniken beschäftigt und zusammen mit anderen Fächern auf die Bewältigung zukünftiger Lebenssituationen in einer Gesellschaft vorbereitet, die zunehmend durch Informationstechnologien und computerbasierte Medien geprägt werden. Wesentliches Ziel des Informatikunterrichts ist die Ausbildung der Sach-, Methoden-, Sozial- und Selbstkompetenz im Umgang mit Informations- und Kommunikationssystemen, welche die Schülerinnen und Schüler zur gleichberechtigten und aktiven Teilhabe am gesellschaftlichen Leben befähigen sowie auf Studium, Beruf und Gestaltung des individuellen Lebens in der Informationsgesellschaft vorbereiten. Die Kompetenzen werden in fachlichen, sozialen und individuellen Bezügen erworben und angewendet. Die im Informatikunterricht erworbenen Kompetenzen befähigen zu zielgerichtetem, situationsangemessenem und verantwortlichem Handeln im Umgang mit Informatiksystemen.

Die zentralen Aufgaben und Ziele des Schulfaches Informatik sind daher:

- Persönlichkeitsentwicklung des Einzelnen durch Förderung seiner Urteils- und Handlungsfähigkeit in der Informationsgesellschaft und die Entwicklung eines verantwortungsbewussten Umgangs mit Informatiksystemen,
- Vermittlung grundlegender Wirkprinzipien von Informatiksystemen und ihrer Beiträge zur Entwicklung von Kultur und Wissenschaft,

- Einordnung der Voraussetzungen, Chancen, Risiken und Folgen bei der Entwicklung zur Informationsgesellschaft und der dabei sich verändernden Lebens- und Arbeitsformen.

Der Beitrag des Informatikunterrichts zur Allgemeinbildung liegt dabei besonders in folgenden Punkten:

### **Analyse, Beschreibungen und Modellierung komplexer Systeme**

Die komplexen Strukturen von Wissenschafts-, Wirtschafts- und Gesellschaftssystemen lassen sich zunehmend nur noch mit Hilfe informatischer Methoden und Verfahren analysieren, beschreiben und beherrschen. Insofern ist die Kenntnis und Verfügbarkeit solcher Methoden für Schülerinnen und Schüler, z. B. Abstraktion, formale Beschreibung und Modellierung von Systemen und die Fähigkeit zur Entwicklung von Lösungsstrategien, ein wesentliches Element für den Aufbau eines zeitgemäßen Weltbildes.

### **Problemlösungsmethoden und ihre Bewertung**

Ausgehend von der Analyse und Modellierung von Systemen ist das algorithmische Problemlösen ein zentraler Bestandteil der Informatik. Die Kenntnis von systematischen Problemlösungsmethoden fördert die Handlungsfähigkeit von Schülerinnen und Schülern in der Informationsgesellschaft, die durch Komplexität und Vernetzung von Systemen geprägt ist. Informatik-Methoden werden damit unverzichtbarer Bestandteil einer Bildung, die Lernende zur Gestaltung der Zukunft befähigen will.

### **Reflexion des Verhältnisses von Mensch und Technik**

Informatikunterricht bietet die Möglichkeit, sich bewusst mit der maschinellen Verarbeitung von Information auseinanderzusetzen. Hierbei sollen auch geschlechtsspezifische Einstellungen thematisiert werden. Das Verhältnis von Mensch und Technik kann in seinem geschichtlichen und gesellschaftlichen Zusammenhang gelernt werden. Dabei sind auch philosophische und historische Fragestellungen von Bedeutung. Es wird die Erkenntnis gefördert, dass ökonomische, ökologische und soziale Zusammenhänge bei der Entwicklung informatischer Lösungen einfließen und diese auch zurückwirken.

### **Verantwortungsbewusster Umgang mit Informatiksystemen**

Die Kenntnis der Möglichkeiten, Grenzen und Gefahren beim Einsatz von Informatiksystemen ist eine Grundlage für die Schülerinnen und Schüler, sich mit normativen und ethischen Fragen, die z. B. den Zugriff auf und die Nutzung von Information sowie den Umgang mit dem Urheber- und Datenschutzrecht betreffen, auseinanderzusetzen.

### **Schöpferisches Denken und Motivation**

Durch Informatiksysteme als Medium und Werkzeug sammeln Schülerinnen und Schüler Erfahrung mit kreativen Gestaltungsmöglichkeiten und eigener, selbstverantworteter Tätigkeit. Der Erwerb von und der Umgang mit Methoden und Verfahren des systematischen Problemlösens trägt dazu bei, schöpferisches Denken bei der Entwicklung eigener Modelle, dem Finden von Lösungsansätzen und beim Transfer auf ähnliche Probleme und Inhalte zu fördern.

### **Kommunikative und kooperative Arbeitsformen**

Schülerinnen und Schüler erfahren bei der Lösung komplexer Probleme, dass Partner-, Team- und Projektarbeit notwendige Voraussetzungen für die Bewältigung der Probleme sind. Die mehrperspektivische Sicht der Informatik setzt Informationsaustausch und Kooperation voraus. Diese Kommunikation wird durch die technischen und medialen Mittel von Informatiksystemen unterstützt bzw. teilweise erst ermöglicht.

## 2 Didaktisch-methodische Grundlagen

### 2.1 Leitlinien

Der Informatikunterricht orientiert sich an vier Leitlinien, die der Gestaltung der Einführungsphase und Qualifikationsphase zu Grunde gelegt werden:

- Umgang mit Informationen,
- Wirkprinzipien von Informatiksystemen,
- informatische Modellierung,
- Wechselwirkung zwischen Informatiksystemen, Individuum und Gesellschaft.

Jede Leitlinie erstreckt sich auf die Kurshalbjahre E1 bis Q4 und wird spiralartig mehrfach durchlaufen, das heißt, in den Kursinhalten aller Kurshalbjahre treten die Ziele und Inhalte mit zunehmender Komplexität und höherem Abstraktionsniveau wieder auf.

#### **Umgang mit Informationen**

Zur Bewältigung der Fülle von Informationen, die uns mittlerweile weltweit zur Verfügung stehen, ist der kompetente Umgang mit Informatiksystemen erforderlich. Die Schülerinnen und Schüler eignen sich daher Handlungsstrategien und Methoden zum Beschaffen, Strukturieren, Darstellen und Präsentieren von Informationen mit geeigneten Informatiksystemen an (z. B. Nutzung produktbezogener Hilfesysteme, elektronischer Nachschlagewerke und Suchmaschinen; hierarchische Strukturierung von Hyperlinks, E-Mail-Adressen oder Dateien, adäquate Auswahl zur Darstellung in Textsystem, Kalkulationsblatt, Datenbank, Pixel- oder Vektorgrafik, eigener Softwarelösung, Präsentation als Website, elektronische Folie, Mindmap, Multimedia-System, Datenbank-Formular oder -Bericht). Sie lernen dabei, in lokalen und globalen Informationsräumen zu navigieren und zu recherchieren, sich selbstständig und kreativ in die Gestaltungsmöglichkeiten mit Informatiksystemen einzuarbeiten und zur Lösung von Problemen adäquate Werkzeuge auszuwählen und anzuwenden. Dabei setzen sie sich mit geeigneten Modellen und Prozessen der Informationsverarbeitung auseinander, erkennen den Unterschied zwischen maschinell verarbeiteten Daten und daraus gewonnenen Informationen und lernen so, Daten zu interpretieren und Informationen auf dem Hintergrund informatischer Aspekte (z. B. Authentizität, digitale Signatur, Web-Adresse) kritisch zu beurteilen.

#### **Wirkprinzipien von Informatiksystemen**

Für das Verständnis und die kompetente Nutzung komplexer Informatiksysteme ist die Kenntnis der zugrunde liegenden Wirkprinzipien sinnvoll. Schülerinnen und Schüler setzen sich daher mit der Struktur von Informatiksystemen auseinander, erkennen die einzelnen Systemkomponenten sowie deren Beziehungen. Sie verstehen die Funktionsprinzipien, nach denen die Systemkomponenten effizient zusammenwirken und wie diese sich in größere Systemzusammenhänge einordnen lassen. Dazu lernen sie grundlegende Ideen und Konzepte (wie z. B. Digitalisierung und Kodierung), die Wirkungsweise wichtiger Komponenten heutiger Informatiksysteme (z. B. Prozessor, Speicher, Netz), Prinzipien, Verfahren und Algorithmen (z. B. Programmierbarkeit, Suchalgorithmen) und den prinzipiellen Aufbau komplexer Basissysteme (z. B. Internet, Datenbanksysteme) kennen.

#### **Informatische Modellierung**

Unter der Leitlinie „informatische Modellierung“ werden die Konzepte und Methoden betrachtet, die für die Konstruktion und die Anwendung von Informatiksystemen in realen Sachzusammenhängen relevant sind. Modellbildung ist ein zentrales Element des Problemlösens mit Informatiksystemen. Modellierung bedeutet dabei im Wesentlichen die Abgrenzung eines für den jeweiligen Zweck relevanten Realitätsausschnittes, die Reduktion auf die wichtigen Merkmale unter Vernachlässigung der unwichtigen sowie die Beschreibung und Strukturierung des Modells mit Hilfe geeigneter Diagramme oder Sprachen. Die Modellierung kann in unterschiedlicher Weise und aus unterschiedlichen Interessen erfolgen; daher sind die Ergebnisse kritisch in Hinblick auf ihre Grenzen und ihre Gültigkeit zu hinterfragen. Das so erhaltene Modell spielt bei der Konstruktion oder Analyse eines Informatiksystems die Rolle eines Bauplans. Auf dieser Basis erfolgt die computergerechte Umsetzung des Modells in ein funktionsfähiges Informatiksystem. Die Schülerinnen und Schüler verstehen dabei, dass jedes Informatiksystem als Kombination von Hard- und Software-Komponenten Ergebnis eines informatischen Modellierungsprozesses ist, das nach seiner Fertigstellung als Bestandteil der realen Welt mit allen Eigenschaften eines unvollständigen, künstlichen Systems wirkt. Sie

können informatische Modellierungstechniken zur Beschreibung von Informatiksystemen und zur Lösung komplexerer Probleme effizient einsetzen. Dabei lernen sie, problemadäquate Software auszuwählen, informatische Problemlösungen sachgerecht zu beurteilen und prinzipielle Grenzen des Problemlösens mit Informatiksystemen zu erkennen.

### **Wechselwirkung zwischen Informatiksystemen, Individuum und Gesellschaft**

Mit der Gestaltung, Entwicklung und dem Einsatz von Informatiksystemen werden Probleme der Lebenswelt gelöst. Aus deren Anwendung ergeben sich Veränderungen in der individuellen Lebensgestaltung oder im gesellschaftlichen Arbeitsumfeld; unter Umständen werden neue Probleme geschaffen. Die soziotechnische Komponente beim Einsatz von Informatiksystemen ist daher Gegenstand des Informatikunterrichts. Die Schülerinnen und Schüler lernen historische Zusammenhänge zwischen gesellschaftlicher und technischer Entwicklung, Voraussetzungen und Folgerungen sowie wirtschaftliche Chancen und Risiken bedeutsamer Anwendungen der Informations- und Kommunikationstechniken kennen. Sie setzen sich mit normativen, ethischen und sozialen Aspekten auseinander, entwickeln einen rechtlichen Orientierungsrahmen und Verantwortungsbewusstsein im Umgang mit moderner Informationstechnik. Aus der Kenntnis der Wirkungen des Einsatzes von Informatiksystemen auf Individuum und Gesellschaft heraus sollen sie Kriterien für menschengerechte Technikgestaltung und deren sozial verträglichen und verantwortungsvollen Einsatz unter Einhaltung der Normen und Werte der demokratischen Gesellschaft entwickeln können.

## **2.2 Arbeitsweisen**

Die aktive Auseinandersetzung mit den Inhalten und Methoden der Informatik im Unterricht erfordert eine intensive Nutzung der schulischen Computeranlagen. Über das Internet können Arbeitsergebnisse zwischen Schule und häuslichem Arbeitsplatz ausgetauscht werden. Die Interaktivität des Mediums Computer ermöglicht in besonderer Weise selbstständiges und selbsttätiges Arbeiten und Lernen der Schülerinnen und Schüler. Damit sind günstige Voraussetzungen zum Lernen nach gemäßigt konstruktivistischer Sicht gegeben, nach welcher Lernen als aktiv-konstruktiver, selbstgesteuerter, situativer und sozialer Prozess verstanden wird. Als Leitkonzept für die Gestaltung geeigneter Lernumgebungen eignet sich die Problemorientierung, die eine Balance zwischen Instruktion und Konstruktion einfordert. Richtungsweisende Gestaltungsprinzipien sind authentische, multiple, soziale und instruktionale Kontexte. Die Themen entstammen daher möglichst realen Problemen mit Anwendungsbezug, die mehrere Sichtweisen zu einem Inhalt erlauben. Soziale Kontexte werden durch Gruppen und Teamarbeit geschaffen. Wo Anleitung und Unterstützung erforderlich ist, muss sie den Lernenden gewährt werden. Problemorientierung im hier beschriebenen Sinn lässt sich sinnvoll mit themen- und projektorientierten wie auch in fächerverbindenden oder fachübergreifenden Unterrichtsphasen umsetzen. Entsprechende Hinweise werden bei der Beschreibung der Kurshalbjahre gegeben.

## **2.3 Grund- und Leistungskurse**

In der Einführungsphase wird Informatik in Grundkursen, in der Qualifikationsphase in Grund- und Leistungskursen unterrichtet. Grund- und Leistungskurse unterscheiden sich nicht grundsätzlich in Zielen und Inhalten. Beide Kursarten zielen auf vertiefte Allgemeinbildung sowie Studier- und Berufsfähigkeit; die Unterschiede liegen im Profil. Grundkurse sind mehr auf Verstehensorientierung, Leistungskurse mehr auf Wissenschaftspropädeutik ausgelegt.

Der Grundkurs vermittelt grundlegende Kenntnisse der fachwissenschaftlichen Methoden und einen Überblick über ihre Anwendung auf wesentliche Bereiche innerhalb und außerhalb der Informatik. Hierzu gehören vor allem Modellbildung, die Problemlösung mit dem Werk- und Denkzeug Computer, die Analyse und Bewertung der Problemlösung sowie die Reflexion der Auswirkungen des Einsatzes von Informatiksystemen. Verstehensorientierung im Grundkurs bedeutet eine eher projekt- und themenorientierte Ausrichtung mit exemplarischer Vertiefung statt Stofffülle. Anwendungsbezüge werden im Grundkurs besonders beachtet, grundlegende Sachverhalte und Strukturen in der Regel punktuell und exemplarisch behandelt, die Motive der teilnehmenden Schülerinnen und Schüler verstärkt aufgenommen. Der Grundkurs ist stärker exemplarisch ausgerichtet; Orientierungswissen hat Vorrang vor Verfügungswissen.

Der Leistungskurs Informatik orientiert sich stärker an der Systematik der Fachwissenschaft und vermittelt ein vertieftes Verständnis und erweiterte Kenntnisse, hat insgesamt also eine wissenschaftspropädeutische Ausrichtung. Wesentliche Begriffe, Konzepte und Strukturen der Informatik werden erarbeitet und angewendet. Dabei spielen formalisiertere Darstellungen bei der Modellbildung, der Problemlösung und Programmierung und ein höherer Grad der Abstraktion der behandelten Inhalte und Begriffe eine wichtige Rolle. Im Leistungskurs wird größerer Wert auf den Erwerb fachlicher Zusammenhänge und Verfügungswissen gelegt. Die Einbettung in einen theoretischen Rahmen schafft Orientierung, stellt die innerfachlichen Bezüge her und ermöglicht den Aufbau von adäquaten Wissensnetzen.

### **3 Umgang mit dem Lehrplan**

#### **3.1 Verbindliche Vorgaben**

Im Teil B sind die Sachgebiete des Informatikunterrichts und ihre Zuordnung zu den Kurshalbjahren der Einführungsphase und der Qualifikationsphase beschrieben. Die Sachgebiete der Kurshalbjahre E1 bis Q3 sind verbindlich. Aus den angegebenen Wahlthemen ist das Sachgebiet des Kurshalbjahres Q4 auszuwählen. Die Zuordnung zu den Kurshalbjahren von E1 bis Q1 ist wegen der inhaltlichen Abhängigkeiten und notwendigen Sequenzialität der Sachgebiete Grundlagen der Programmierung und Objektorientierte Modellierung verbindlich. Die Sachgebiete Datenbanken und Theoretische Informatik können in ihrer Reihenfolge ausgetauscht werden.

Die verbindlichen Unterrichtsinhalte der Sachgebiete werden durch Stichworte ergänzt und präzisiert. Die Reihenfolge der Unterrichtsinhalte ist nach sachlogischen Zusammenhängen beschrieben. Sie kann gemäß den Anforderungen aus dem gewählten Zugang zum Sachgebiet variiert werden. Nicht alle Unterrichtsinhalte und Stichworte müssen in gleicher Intensität behandelt werden. Je nach didaktischer und methodischer Planung können Schwerpunktsetzungen vorgenommen werden.

Gemäß den Vorgaben der Oberstufen- und Abiturverordnung (OAVO) werden die Grundkurse in der Einführungsphase zweistündig, in der Qualifikationsphase in der Regel dreistündig unterrichtet. Zwei Drittel der gesamten zur Verfügung stehenden Unterrichtszeit sind für die Behandlung der verbindlichen Unterrichtsinhalte vorgesehen, ein Drittel für die fakultativen Unterrichtsinhalte. Wird in einem Grundkurs der Qualifikationsphase zweistündig unterrichtet, so ist die gesamte Unterrichtszeit für die verbindlichen Unterrichtsinhalte vorgesehen, die fakultativen Unterrichtsinhalte entfallen. Die ausgewiesenen Stundenansätze beziehen sich auf verbindliche Unterrichtsinhalte. Bei den fakultativen Unterrichtsinhalten wurde auf eine Unterscheidung zwischen Grund- und Leistungskursen zugunsten einer breiteren Auswahl verzichtet.

Die im unterrichtspraktischen Teil des Lehrplans angegebenen Themen orientieren sich an der Struktur der Wissenschaft Informatik. Aus der Verwendung relevanter Fachbegriffe in den Unterrichtsinhalten und Stichworten darf nicht abgeleitet werden, dass sie wie in der Fachwissenschaft behandelt werden können. Den Zielsetzungen eines allgemein bildenden Unterrichts wird nur eine schüler- und unterrichtsgemäße Behandlung der Unterrichtsinhalte gerecht; übertriebene fachwissenschaftliche Ansprüche sind zu vermeiden.

Im Leistungskurs lernen die Schülerinnen und Schüler mindestens eine weitere Programmiersprache mit prädikativem oder funktionalem Sprachparadigma kennen. Der Einsatz von Prolog wird durch geeignete Unterrichtsmaterialien auf dem Hessischen Bildungsserver unterstützt.

### 3.2 Lern- und Prüfungsbereiche

Unbeschadet der didaktischen und methodischen Freiheit bei der Schwerpunktsetzung und Intensität der Behandlung einzelner Teilgebiete ist darauf zu achten, dass die folgenden Lern- und Prüfungsbereiche für die Abiturprüfung zur Verfügung stehen:

#### 1. Algorithmusbezogene Qualifikationen

Algorithmische Grundlagen, Gegenstände und Methoden der Informatik

- Algorithmen und Datenstrukturen
- Modellierung
- Methoden der Software-Entwicklung

#### 2. Rechnerbezogene Qualifikationen

Funktionsprinzipien von Hard- und Softwaresystemen

- Programmiersprachen und -umgebungen
- Rechnermodelle und reale Rechnerkonfigurationen
- Theoretische Grundlagen

#### 3. Anwendungsbezogene Qualifikationen

Anwendung von Hard- und Softwaresystemen und deren gesellschaftliche Auswirkungen

- Anwendungsgebiete
- Mensch-Maschine-Schnittstelle
- Grenzen und Möglichkeiten, Chancen und Risiken des Einsatzes der Informations- und Kommunikationstechniken

**Teil B      Unterrichtspraktischer Teil  
Der Unterricht in der Sekundarstufe II****Übersicht der verbindlichen Themen**

<b>Lfd. Nr.</b>	<b>Verbindliche Unterrichtsthemen</b>	<b>Stundenansatz</b>
E1	Internet	23
E2	Grundlagen der Programmierung	23
Q1	Objektorientierte Modellierung	GK 36/LK 63
Q2	Datenbanken	GK 36/LK 63
Q3	Konzepte und Anwendungen der Theoretischen Informatik	GK 36/LK 63
Q4	Wahlthema	GK 24/LK 43

## **1 Die verbindlichen und fakultativen Unterrichtsinhalte der Einführungs- und der Qualifikationsphase**

### **1.1 Die Einführungsphase (E1 und E2)**

Von der Sekundarstufe I bringen Schülerinnen und Schüler unterschiedliche Vorkenntnisse über wichtige Anwendungsgebiete der Informations- und Kommunikationstechniken mit. In den Lehrplänen sind die jeweiligen Beiträge der Fächer zur informations- und kommunikationstechnischen Grundbildung und die Einsatzmöglichkeiten des Computers im Fachunterricht ausgewiesen. Aufbauend auf den bisherigen Erfahrungen und Kenntnissen im Umgang mit dem Computer ist es Ziel der Einführungsphase, die unterschiedlichen inhaltlichen Voraussetzungen der Schülerinnen und Schüler einander anzugleichen, einen Überblick über das Fach Informatik zu liefern und inhaltliche und methodische Grundlagen für das Fach Informatik in der Qualifikationsphase zu schaffen.

Viele Schülerinnen und Schüler haben in der Einführungsphase erstmalig Gelegenheit, sich mit informatischen Fragestellungen auseinander zu setzen, die über den Umgang mit dem Computer und dessen Benutzung hinausgehen. Dem Informatikunterricht kommt deshalb eine wichtige Orientierungsfunktion mit maßgeblicher Bedeutung für die Fortführung des Faches Informatik in der Qualifikationsphase zu. Im Unterricht muss daher ein Bild der Informatik vermittelt werden, das den aktuellen Gegebenheiten entspricht, zur sachkompetenten und verantwortungsbewussten Nutzung von Informatiksystemen hinführt und den Weg zum Verständnis und zur Beurteilung weiterer Entwicklungen in diesem Bereich ermöglicht. Dies soll erreicht werden, indem nicht nur verschiedene Inhalte aus dem Gebiet der Informatik, sondern explizit auch die hier angesiedelten unterschiedlichen Arbeitsformen und Problemlösungsstrategien vorgestellt und angewendet werden. Am Ende des Schuljahres kann dann eine aus eigener Erfahrung begründete Entscheidung getroffen werden, dieses Fach als Grund- oder Leistungskurs weiterzubetreiben.

Schülerinnen und Schüler haben möglicherweise schon im Wahlpflicht- oder Wahlunterricht Informatik der Sekundarstufe I die in der Einführungsphase zu behandelnden Themen Internet und Grundlagen der Programmierung kennen gelernt. Im Sinne eines Spiralcurriculums werden dann diese Themen in der Einführungsphase vertieft.

## 1.1.1 E1

E1

Internet

Std.: 23

**Begründung:**

In einer vernetzten Welt gilt es nicht nur Techniken zu erlernen, die zur Beherrschung eines Informationssystems wie des Internets notwendig sind. Genauso wichtig ist es, sich die Auswirkungen bestimmter Vorgänge auf einen oft komplexen Gesamtzusammenhang zu vergegenwärtigen und die vielfältigen Verbindungen und Abhängigkeiten des betrachteten Objektes zu und von seiner Umgebung zu erkennen. Isolierte Abläufe gibt es in einer solchen Welt kaum noch jede Handlung erzeugt als Reaktion eine Kette von Veränderungen, von denen viele nicht offensichtlich und damit meist unerwartet sind. Hier sind demnach Repräsentations- und Dokumentationstechniken notwendig, welche zum einen über die sonst in diesem Bereich übliche lineare Struktur hinausgehen und zum anderen der Vielfalt der Informationssysteme dadurch Rechnung tragen, dass sie vom Ausgabemedium weitgehend unabhängig sind. Der sinnvolle Umgang mit einer anwachsenden Informationsflut ist eine wichtige Qualifikation in der heutigen Gesellschaft. Hierzu kann der Informatikunterricht durch Entwickeln von geeigneten Strategien zum Herausfiltern der relevanten Informationen beitragen. HTML ist der ideale Ansatzpunkt, um sich in diesem Problemfeld zu bewegen.

Die modernen Informations- und Kommunikationstechnologien haben unsere Gesellschaft grundlegend verändert und werden dies auch zukünftig tun. Der gesellschaftliche Wandel bedeutet auch konkrete Veränderungen für den Einzelnen. Man betrachte nur einmal heutige Kommunikationsstrukturen, -formen und -wege. Eine fundierte Kenntnis des Internets mit seinen Möglichkeiten und Diensten ist das Ziel des Unterrichts, wobei Vorerfahrungen koordiniert, Vorwissen in den richtigen Kontext eingeordnet und Zusammenhänge wie Grundlagen erklärt werden. Zumindest exemplarisch soll eine nähere Untersuchung der zu Grunde liegenden Strukturen und Mechanismen durchgeführt werden. Die Schülerinnen und Schüler lernen auf diese Weise die Wirkprinzipien des umfassenden Informationssystems Internet kennen.

Adressen und Protokolle bilden das Fundament des Internets. Durch die einheitliche IP-Adressvergabe wird die globale Kommunikation ermöglicht, die über standardisierte Protokolle abgewickelt wird. Ein Protokoll in Funktion und Einsatz kann Gegenstand einer näheren Untersuchung sein, z. B. POP3 oder SMTP als Basis der Kommunikation per E-Mail.

Das Verständnis von Client-Server-Strukturen und deren Auswirkungen ist unverzichtbar. An einem Beispiel kann die Realisierung der Prinzipien untersucht werden – News oder E-Mail bieten sich an. Dabei können Sicherheitsaspekte, Rechte und Benutzerverwaltung den Schülerinnen und Schülern anhand geeigneter Beispiele auch aus dem gesellschaftlichen und volkswirtschaftlichen Bereich vermittelt werden.

Fakultativ bietet sich die tiefer gehende Behandlung von *Suchmaschinen* im Unterricht an. Dabei sollten sowohl die Meta-Tags und ihre Verwendung als auch die Arbeitsweise von Suchmaschinen selbst unter vielfältigen Gesichtspunkten untersucht werden. Dies betrifft dann auch das Gebiet der Datenbanken, die im Internet eine immer größer werdende Rolle spielen. Auf deren weiterführende Behandlung in Q2 sei hier schon hingewiesen. Die Grundlagen der Scriptsprachen (z. B. Perl oder PHP) bieten einen alternativen Zugang zur gleichen Problematik.

**Verbindliche Unterrichtsinhalte/Aufgaben:**

Repräsentation per HTML	Ein Informationssystem auf Hypertext-Basis mit HTML erstellen können, auch mit Frames und Formularen
Internet	Struktur, Dienste, individuelle und gesellschaftliche Bedeutung, Urheber- und Datenschutzrecht
Adressen und Protokolle	IP-Adressen, DNS, TCP/IP als Protokoll computervermittelter Kommunikation
Client-Server-Architektur	Aufbau des Systems: Client, Server, Dienst, Protokoll, Sicherheitsaspekte (SSL), Internet, Datei-Server, Rechte, Benutzerverwaltung

**Fakultative Unterrichtsinhalte/Aufgaben:**

Serverdienste und Anwendungsprotokolle	Exemplarisches Untersuchen, z. B. eines News-, POP3- oder SMTP-Servers und des Protokolls
Funktionsweisen von Suchmaschinen	Indizes, Einblick in Datenbanken
Meta-Tags	Steuern von Suchmaschinen, Meta-Information als Qualitätskriterium
Internet-Datenbanken	Dynamische Webseiten, Zugriff über Scriptsprachen

**Arbeitsmethoden der Schülerinnen und Schüler/Hinweise und Erläuterungen:**

Die gründliche Kenntnis und Beherrschung des Informationssystems Internet führt zu einer für Beurteilungen und Entscheidungen notwendigen Kompetenz. Diese unterstützt dann sowohl das Planen als auch das verantwortliche Handeln und führt schließlich zu möglichen Lernfortschritten in kommunikativen wie sozialen Bereichen. Eine sich in diesem Halbjahr anbietende Projektarbeit in Kleingruppen unterstützt das Erreichen von Lernzielen aus Gebieten wie der Teamfähigkeit und dem selbstständigen Arbeiten ebenso wie Ansätze im fachübergreifenden und fächerverbindenden Vorgehen.

Das Thema HTML bietet geeignete Zugänge zum projektorientierten Arbeiten, die genutzt werden sollen. So lassen sich auch die schon bei einigen Schülerinnen und Schülern sicherlich vorhandenen Vorkenntnisse durch geeignete Vertiefungen integrieren. Die Schülerinnen und Schüler erstellen arbeitsteilig Informationsseiten, die auf der Grundlage der zuvor modellierten Hypertextstruktur mit Verweisen zu einem vernetzten Informationssystem zusammengefasst werden. Dazu müssen sie Vereinbarungen zum Beispiel zur Namensgebung, Ordnerstruktur und zum Layout treffen, die sie zur erfolgreichen Koordination einhalten müssen. Das Projekt vereint sowohl Wissensgewinnung als auch Wissensrepräsentation in einem Hypertextsystem mit lokalen und globalen Verweisen und verdeutlicht so den vernetzten Charakter des Internets.

Sinnvoll ist ein fächerverbindender Ansatz, der die inhaltliche Arbeit in einem weiteren Unterrichtsfach mit den beschriebenen Vorgehensweisen und Methoden der Informatik vereint. Die in einer Recherchephase im Informatikunterricht gefundenen Informationen werden im Unterricht des angebundenen Faches gesichtet und bewertet. Die Qualität der gefundenen Information kann auf der Basis informatischer Aspekte, wie zum Beispiel URL und Meta-Tags, wie auch fachlicher Gesichtspunkte beurteilt werden. Die Repräsentation der im Fachunterricht erarbeiteten Ergebnisse als Hypertextsystem findet anschließend im Informatikunterricht statt.

**Querverweise:****Massenmedien und Kultur:**

D, E, F, Spa, Ita, L, PoWi

**Programmierung und Simulation:**

M, Ch, Phy, PoWi

**Internet und Hypertext:**

Bio, PoWi

**Berücksichtigung von Aufgabengebieten  
(§ 6 Abs. 4 HSchG):**

Rechtserziehung

## 1.1.2 E2

E2

Grundlagen der Programmierung

Std.: 23

**Begründung:**

Die Schülerinnen und Schüler haben im Fachunterricht der Sekundarstufe I vielfältige Anwendersysteme kennen gelernt, damit konkrete Anwendungsprobleme gelöst und individuelle sowie gesellschaftliche Folgen der Nutzung reflektiert. Kennzeichnend für den Umgang mit Anwendersystemen ist die interaktive Arbeit, d. h. das benutzte System führt direkt die Aktionen aus, die der Anwender per Tastatur oder Maus auslöst.

Für den kompetenten Umgang mit modernen Informatiksystemen benötigt man aber auch Hintergrundwissen über Wirkprinzipien und Möglichkeiten automatischer Informationsverarbeitung. Der Zugang zu diesen Inhalten erfolgt über das Thema *Grundlagen der Programmierung*. Da der Begriff der Programmierung sehr unterschiedlich interpretiert wird, ist eine Klärung nötig. Programmierung wird nicht als ingenieurwissenschaftliche Tätigkeit, sondern ebenso wie das elementare Rechnen in der Mathematik als eine Primärerfahrung mit der Informatik verstanden. Zentrale Begriffe der Informatik erwachsen aus den Erfordernissen des Programmierens; die Erfahrung des Programmierens spielt eine Schlüsselrolle für das Verständnis informatischer Grundbegriffe. Programmierung wird aufgefasst als Problemlösen (Modellieren und Strukturieren) unter Anwendung von Informatikprinzipien und -methoden, als Teilgebiet der Informatik, das die Methoden und Denkweisen beim Entwickeln von Programmen umfasst. Die Programmiersprache ist lediglich Mittel zum Zweck und steht nicht im Zentrum des Unterrichts. Ein Programmierkurs, der von Sprach- anstatt von Problemstrukturen ausgeht, verfehlt die Zielsetzungen eines allgemein bildenden Unterrichts.

Programmieren beginnt mit dem Modellieren der betrachteten Problemsituation. Dabei werden für den jeweiligen Zweck die wichtigsten Merkmale und Aktionsmöglichkeiten der beteiligten Objekte unter Vernachlässigung der unwichtigen herausgearbeitet, beschrieben und strukturiert. Daraus ergibt sich das informatische Modell, das die Rolle eines Bauplans für die Konstruktion eines Informatiksystems spielt. Bei der Modellierung sind äußere Aspekte in Form der Gestaltung der Benutzungsoberfläche zur Ein- und Ausgabe und innere Aspekte wie Darstellung von Informationen durch Daten sowie Verarbeitung von Daten durch Algorithmen zu beachten. Algorithmen können mit Hilfe von Struktogrammen visualisiert werden. Weitere wichtige Hilfsmittel sind Zustands- bzw. Ablaufdiagramme.

Die Konstruktion des informatischen Modells erfolgt mit den Mitteln einer objektorientierten Programmiersprache. Um an die Erfahrungswelt der Schülerinnen und Schüler anzuschließen, ist der Einsatz eines Entwicklungssystems erforderlich, mit dem Programme für grafikorientierte Benutzungsoberflächen entwickelt werden können. Insbesondere empfehlen sich hierzu Delphi oder Java. Diese beiden Entwicklungssysteme werden durch entsprechende Beispiele, Unterrichtsvorschläge und Materialien auf dem Hessischen Bildungsserver unterstützt. Sie beinhalten ein umfangreiches Hilfesystem, das zur Förderung des selbstständigen Lernens eingesetzt werden kann. Ein zusätzlicher Beitrag zur Stärkung des Sprachenlernens erfolgt durch die Nutzung der englischsprachigen Programmdokumentationen.

An die erfolgreiche Implementierung eines Informatiksystems schließt sich deren Präsentation, die Diskussion der Lösung und eventueller offener Probleme samt ihrer individuellen und gesellschaftlichen Bedeutung sowie die kritische Reflexion des gesamten Lösungsprozesses an. Damit erfahren die Schülerinnen und Schüler ihre Fortschritte in Bezug auf Fach-, Methoden-, Sozial- und Selbstkompetenz.

**Verbindliche Unterrichtsinhalte/Aufgaben:**

Variablen	Als benannter Behälter für Werte eines bestimmten Datentyps, Wertzuweisung
Einfache Datentypen mit deren relevanten Operationen und Relationen	Integer, Real, Char, Boolean
Strukturierte Datentypen mit ihren relevanten Operationen und Relationen	String, Feld (array)
Kontrollstrukturen	Anweisungen, Sequenzen, Schleifen, Fallunterscheidungen, Syntaxdiagramme
Struktogramme	grafische Darstellung von Algorithmen
Modularisierung	Prozeduren, Parameter
Benutzeroberfläche Mensch-Maschine-Interaktion	grundlegende Ein-/Ausgabe-Komponenten Ereignisse, Ereignisroutinen
Zustandsorientierte Modellierung	Zustände, Übergänge, Zustandsdiagramme

**Fakultative Unterrichtsinhalte/Aufgaben:**

Externes Speichern von Zuständen	Textdateien
Funktionale Modellierung	Aufteilung komplexer Systeme in Teilsysteme, Datenflussdiagramme, schrittweise Verfeinerung, Schnittstellen

**Arbeitsmethoden der Schülerinnen und Schüler/Hinweise und Erläuterungen:**

Die Unterrichtsinhalte werden anhand von Themen erarbeitet. Ein themenorientierter Zugang ist an Verstehensorientierung und auf die Initiierung und Stärkung sachbezogener Motivation ausgerichtet. Er erlaubt exemplarische Vertiefung und umgeht somit das Problem der Stofffülle. Die Themen sind so auszuwählen, dass sie einen Beitrag zur Stiftung kultureller Kohärenz, zur Weltorientierung und zur Entfaltung von Verantwortungsbereitschaft leisten. Sie sollten sich auf authentische Problemsituationen beziehen, die auf Grund ihres Realitätsgehalts und ihrer Relevanz dazu motivieren, neues Wissen oder neue Fertigkeiten zu erwerben. Beispiele für geeignete Themen sind Kalender, Kryptografie, Grafik, Spiele, Ton- oder Bildverarbeitung, Auswertung von Messergebnissen, Simulation technischer Automaten. Auf Grund des interdisziplinären Charakters der Informatik gibt es viele Bezüge zu den anderen Unterrichtsfächern und damit Gelegenheit, auch fachübergreifende und fächerverbindende Themen zu behandeln.

Im Sinne des problemorientierten Lernens sollen Phasen expliziter Instruktion durch die Lehrkraft und konstruktiver Aktivität der Lernenden in einer sinnvollen Balance stehen. Die Schülerinnen und Schüler müssen ausreichend Gelegenheit haben, praktische Erfahrungen im Umgang mit dem Programmentwicklungssystem zu erwerben und Möglichkeiten des selbstständigen Lernens zu nutzen. In der Einführungsphase muss von sehr heterogenen Vorkenntnissen und großen Differenzen in den Lern- und Arbeitstechniken ausgegangen werden. Problemorientiertes Lernen bietet unter diesen Umständen sinnvolle Voraussetzungen für die innere Differenzierung sowie für Partner- und Teamarbeit.

Mit der zustandsorientierten Modellierung werden typischerweise endliche Automaten beschrieben, die genauer in Q3 behandelt werden. Da Variablen im Grunde nichts anderes als Programmzustände beschreiben und die Wertzuweisung in imperativen Programmiersprachen im engeren Sinne eine Zustandsänderung darstellt, ist die Verwendung von Variablen immer als zustandsorientierte Modellierung zu verstehen

und daher verbindlicher Unterrichtsinhalt dieses Kurshalbjahres. Beispiele für die zustandsorientierte Modellierung sind auf dem Hessischen Bildungsserver dokumentiert (Ampelschaltung, POP3- Protokoll).

Am Ende der Einführungsphase kann ein kleines Projekt dazu dienen, in Teamarbeit und arbeitsteiliger Vorgehensweise bei der Analyse von Problemen, der Erstellung und dem Test von Lösungen sowie der Dokumentation des Lösungsprozesses einzuführen.

---

**Querverweise:**
**Massenmedien und Kultur:**

D, E, F, Spa, Ita, L, PoWi

**Programmierung und Simulation:**

M, Ch, Phy, PoWi

**Internet und Hypertext:**

Bio, PoWi

**Berücksichtigung von Aufgabengebieten  
(§ 6 Abs. 4 HSchG):**

Rechtserziehung

---

**1.2 Die Qualifikationsphase (Q1 bis Q4)****1.2.1 Q1**

<b>Q1</b>	<b>Objektorientierte Modellierung</b>	<b>Std.:</b> <b>GK 36</b> <b>LK 63</b>
-----------	---------------------------------------	----------------------------------------------

**Begründung:**

In der Einführungsphase wird mit den Grundlagen der Programmierung ein Fundament geschaffen, auf dem in Q1 das Thema objektorientierte Modellierung aufbaut. In Q1 befassen sich die Schülerinnen und Schüler mit Konzepten, Methoden und Verfahren der objektorientierten Modellbildung in den Phasen Analyse, Design und Programmierung. Die objektorientierte Analyse ist Ausgangspunkt der Modellierung. Ein relevanter Ausschnitt der realen Welt wird untersucht und unter Vernachlässigung der unwichtigen auf die bedeutsamen Merkmale reduziert. Der Abstraktionsprozess liefert Attribute, Methoden, Objekte und Klassen, welche in der Unified-Modeling-Language (UML) dargestellt werden. Mittels objektorientiertem Design wird die Struktur eines passenden Informatiksystems entworfen. Strukturelle Beziehungen, zeitliche Abläufe und Datenrepräsentierung sind relevante Gestaltungsaufgaben, deren Lösung sich in Konstruktionsplänen dokumentiert. Die geplante Struktur des Informatiksystems wird mit der gewählten Programmiersprache in ein lauffähiges Programm übersetzt, mit dem die Lösung getestet werden kann.

Es schließt sich eine kritische Reflexion der Vorgehensweise und der erzielten Lösung an. Dabei spielen Auswirkungen des Einsatzes dieses Informatiksystems eine wichtige Rolle. Exemplarisch kann daran aufgezeigt werden, dass Informatiksysteme soziotechnische Systeme sind, die auf die Lebens- und Arbeitswelt der Betroffenen Einfluss haben.

In Bezug auf Algorithmen findet eine Vertiefung statt. Die effiziente Informationsverarbeitung erfolgt auf der Basis von Standardalgorithmen. Die Rekursion als wichtiges Programmierkonzept wird anhand von Beispielen mit der Iteration verglichen. Beispiele aus dem Bereich der Grafik ermöglichen eine unmittelbare Rückmeldung über die Korrektheit der Lösung. Im Grundkurs genügt die Behandlung elementarer Algorithmen zum Suchen und Sortieren. Im Leistungskurs sind auch effiziente Algorithmen nach dem Prinzip *Teile und Herrsche* bzw. *Backtracking* zu behandeln.

Die Verarbeitung großer Datenmengen und flexibler Datenstrukturen erfordert die Behandlung abstrakter Datentypen. Im Sinne objektorientierter Modellierung entwirft und implementiert man abstrakte Datentypen als Klassen. In Attributen werden die Daten gespeichert; Standardalgorithmen erscheinen als Methoden der jeweiligen Klasse.

Moderne Programmentwicklungssysteme bieten fertige Klassen für die zu betrachtenden abstrakten Datentypen an, so dass diese nicht selbst entwickelt werden müssen. Unter diesem Aspekt spielen eher die Orientierung in der zum Entwicklungssystem gehörenden Klassenhierarchie, die Fähigkeit, die für die eigene Problemlösung relevanten Klassen und Methoden zu finden, und die Anwendung durch Ableitung spezialisierter Klassen, z. B. für spezielle Datentypen, eine wichtige Rolle.

Demgegenüber spielt unter dem Aspekt der eigenen Konstruktion das Zeigerkonzept zur Realisierung dynamischer Datenstrukturen eine wichtige Rolle. Die explizite Arbeit mit Zeigern ist nicht unbedingt erforderlich. Ein Einblick in das Zeigerkonzept als Variablen mit dem Variablentyp Adressen genügt. Die abstrakten Datentypen Keller und Schlange können beispielsweise als spezielle lineare Listen aufgefasst und daher von einer vorgegebenen Listen-Klasse abgeleitet werden.

**Verbindliche Unterrichtsinhalte/Aufgaben:**

Objektmodell	Identifikation von Objekten Objekt als Exemplar einer Klasse Kommunikation über Botschaften
Klassen	Attribute als Datenstruktur zur Repräsentierung der Information über ein Objekt Methoden als Schnittstellen für den Zugriff auf Attribute und zum Nachrichtenaustausch Darstellung von Klassen, Objektbeziehungen und Vererbung mit der grafischen Modellierungssprache UML Vererbung und Klassenhierarchie
Standardalgorithmen	rekursive und iterative Verfahren einfache Such- und Sortierverfahren binäre Suche (LK)
Abstrakte Datentypen (LK, GK fakultativ)	Repräsentierung und Standardoperationen Keller, Schlange lineare Liste, binärer Suchbaum
Effiziente Algorithmen (LK, GK fakultativ)	schnelle Sortierverfahren Suchen durch Backtracking Teile- und Herrsche-Prinzip
Komplexität von Algorithmen (LK, GK fakultativ)	polynomiale und exponentielle Zeitkomplexität
Grundkonzepte des Software-Engineerings	Problemanalyse, Modellierung, Entwurf, Implementation, Test, Revision, Dokumentation

**Fakultative Unterrichtsinhalte/Aufgaben:**

Abstrakter Datentyp	Abbildungen aus (Schlüssel-, Wert-)Paaren Hash-Tabelle
Grafen	Datenstruktur Tiefen- und Breitensuche Grafenalgorithmen
Internetprogrammierung	Sockets, Client-Server-Konzept, z. B. POP3-Explorer, Mikro-Browser
Heuristische Verfahren	z. B. Bewertung von Spielsituationen
Polymorphie	z. B. Manipulation grafischer Objekte
Suche in Texten	Volltextsuche, Mustererkennung

**Arbeitsmethoden der Schülerinnen und Schüler/Hinweise und Erläuterungen:**

Die Schülerinnen und Schüler wenden im Laufe des Kurshalbjahres die Methodik der objektorientierten Modellierung auf ein Problem an, das auf ihr Leistungsvermögen und den zur Verfügung stehenden Zeitrahmen abgestimmt ist. Je nach Erfahrung mit Projektarbeit arbeiten sie in Teams an verschiedenen Projekten oder gemeinsam an einem Projekt. Die Bearbeitung fachübergreifender Probleme bietet sich an, wobei möglichst die behandelten abstrakten Datentypen und Algorithmen zum Einsatz kommen sollen. Die Dokumentation des Projekts erfolgt sinnvollerweise in Form von HTML-Dokumenten (siehe Grundkurs E1). Bei Verwendung von Java kann das Projektergebnis als Applet direkt in die Dokumentation eingebunden werden. Die Vorgabe einer Struktur für die Dokumentation ist hilfreich. Die erforderlichen UML-Diagramme können mit Grafikwerkzeugen hergestellt werden. Die Qualität der Dokumentation ergibt sich unter anderem durch die Klarheit im Aufbau, die Konzentration auf das Wesentliche, die verständliche und fachlich einwandfreie Beschreibung, den expliziten Bezug zwischen Text, Bild und Diagrammen, die kritische Diskussion der Ergebnisse und die Reflexion der Auswirkungen beim Einsatz des Informatiksystems.

Zum Abschluss der Projektphase stellen die Teams ihre Arbeit in Form einer Präsentation im Plenum vor. Die Präsentation hat in Bezug auf die Dokumentation ihren eigenen Stellenwert, der sich aus einem Vortrag in Kombination mit Präsentationsfolien oder HTML-Seiten ergibt. Für die Beurteilung der Präsentation sind auch kommunikative Fähigkeiten, angewandte Präsentationstechnik und bewusste Körpersprache heranzuziehen.

**Querverweise:****Modellierung:** PoWi, Ch**Berücksichtigung von Aufgabengebieten  
(§ 6 Abs. 4 HSchG):**

## 1.2.2 Q2

Q2	Datenbanken	Std.: GK 36 LK 63
----	-------------	-------------------------

**Begründung:**

Datenbanksysteme sind heute mit die wichtigsten Anwendungsgebiete von Computern in Wirtschaft, Verwaltung, Wissenschaft, Technik und Freizeit. Im Hinblick auf zunehmende weltweite Vernetzung und den damit verbundenen Zugriff auf eine fast unendliche Flut von Daten rücken die Organisation dieser Daten und die Operationen mit ihnen immer mehr in den Mittelpunkt. Grundlagen solcher Systeme sind Datenbanken, in denen Texte, Bilder und andere multimediale Elemente sowie weitere Informationen unterschiedlichster Art in digitaler Form gespeichert werden. Relevante Fragestellungen im Zusammenhang mit Datenbanksystemen betreffen die Strukturierung, Darstellung und Codierung der Daten, die Informationsspeicherung allgemein, die Probleme der Wiedergewinnung, der Selektion und der Filterung von Informationen sowie natürlich auch Aspekte der Sicherheit und des Schutzes der Daten vor fehlerhaftem oder unzulässigem Zugriff.

Im Grund- und Leistungskurs gilt es deswegen, die Grundlagen von Datenbanksystemen zu vermitteln und ein relationales Datenbankmodell zu behandeln. Die Modellierung eines Ausschnittes der realen Welt durch Abstraktion unbedeutender Details und Reduktion auf die relevanten Objekte und Zusammenhänge auf der Basis von Entity-Relationship-(ER)-Diagrammen schärft den Blick für das Wesentliche und ermöglicht so, in der zur Verfügung stehenden Zeit Problemlösungen zu realisieren, die der Erfahrungswelt der Schülerinnen und Schüler entstammen. Die Relationenalgebra bezieht man sinnvollerweise auf die bekannten Operationen der Mengenlehre. Darauf basierend lassen sich Operationen wie zum Beispiel der *Join* leicht einführen.

Die Umsetzung und Bearbeitung der ER-Modelle (ERM) soll realitätsnah in einem modernen Datenbanksystem erfolgen, das auf der Basis der definierten Datenbeziehungen auch unterschiedliche Datensichten erzeugen und diese durch Abfragen zum Beispiel mit Query-By-Example (QBE) gezielt auswerten kann. Auch im Grundkurs kann auf die plattformunabhängige Standard-Abfragesprache Structured-Query-Language (SQL) nicht verzichtet werden.

**Verbindliche Unterrichtsinhalte/Aufgaben:**

Entity-Relationship-Modell	Objekte, Beziehungen und Kardinalitäten (1:1, 1:n, n:m)
Abbildung des Entity-Relationship-Modells in das Relationenmodell	Schlüssel, Optimierung bei obligatorischen Beziehungen
Normalisierung (LK, GK fakultativ)	1. bis 3. Codd'sche Normalform
Realisierung in einem Datenbanksystem	Datenfeld, Datensatz, Tabelle, Beziehungen
Relationenalgebra	Selektion, Projektion, Join
3-Schichtenmodell (LK)	interne, konzeptuelle und externe Schicht
Abfragen mit QBE und SQL	interaktive Abfragen mit QBE, Abfragen mit SQL-Select-Befehlen
DDL, DML, DCL (LK, GK fakultativ)	Erweiterung des SQL-Sprachumfangs
Datensicherheit (LK)	Benutzerprofile im RDBMS, Transaktionsverwaltung

Datenschutz

Datenschutzgesetz  
Recht auf informationelle Selbstbestimmung**Fakultative Unterrichtsinhalte/Aufgaben:**Zugriff auf Datenbanken  
mit einer Programmiersprache

Datenbanktreiber und -schnittstellen

Datenbanken im Netz

SQL-Server, Datenbankzugriff über das Internet

Wissensbasierte Systeme

Datenbanken mit Prolog

**Arbeitsmethoden der Schülerinnen und Schüler/Hinweise und Erläuterungen:**

Die Erstellung einer *Miniwelt* steht im Mittelpunkt der Modellierung. Das ERM ist zwar sehr leistungsfähig und gibt den Schülerinnen und Schülern eine solide Basis für die Konstruktion der eigentlichen Datenbank, die Abstrahierung aber ist ein wichtiger Prozess, der an einigen Beispielen geübt und durch die Erarbeitung z. B. von *Geschäftsregeln* und deren Umsetzung unterstützt werden muss. Eine konsequente Umsetzung des ERM in Tabellen sowie einige schematisierte Betrachtungen zur Kardinalität müssen im Leistungskurs durch eine Betrachtung der Normalisierung und die entsprechenden Codd'schen Normalformen ergänzt werden.

Im Zusammenhang mit Fragen der Datensicherheit muss im Leistungskurs einer der beiden Aspekte *Benutzerprofile im DBMS* oder *Transaktionskontrolle* behandelt werden.

Die Anwendung der erlernten Modelle, Konzepte und Verfahren erfolgt an einem Datenbankprojekt. Dies kann aus schulischen Bedürfnissen (z. B. Projektwoche, Sportfest, Bücherei) oder Anforderungen aus einem anderen Fach erwachsen (z. B. Literaturdatenbank, Personalinformationssystem, geographische Datenbank).

Der Zugriff auf eine Datenbank mit der Programmiersprache verbindet in sinnvoller Weise die Kursinhalte aus Q1 und Q2. Mit ihr können Anwendungen entwickelt werden, welche über die SQL-Schnittstelle auf die Datenbank zugreifen. *Rapid-Prototyping* lässt sich mit den GUI-Elementen der Entwicklungsumgebung als verbreitete Methode der Anwendungsentwicklung anhand eines solchen Beispiels leicht und unmittelbar erfahren.

Im Leistungskurs ist es sinnvoll, eine Scriptsprache einzusetzen und die Begriffe *Vernetzung – Client-Server-Strukturen – HTML – Abfragesprache – Scriptsprache* in einen gemeinsamen Kontext mit den anderen Aspekten der Datenbanken zu stellen. Die Auseinandersetzung mit Datenbanken im Internet schließt an die Inhalte des Kurses Internet aus der E1 an. Die Schülerinnen und Schüler erhalten Einblick in die Wirkungsweise datenbankbasierter Client-Server-Systeme, welche die Basis dynamischer Webinhalte bilden. Sie setzen ihre Kenntnisse in einem kursübergreifenden Projekt ein, in dem über Webformulare Datenbanken im Intranet oder Internet abgefragt oder bearbeitet werden.

Das fakultative Thema *Datenbanken mit Prolog* bietet einen sinnvollen Einstieg in das im Leistungskurs erforderliche zweite Programmiersprachen-Paradigma. Die Grundelemente von Prolog werden dabei soweit eingeführt, dass Abfragen auf Prolog-Datenbanken durchgeführt werden können.

**Querverweise:****Datenbanken:** PoWi, G, Ek, Ch, M**Berücksichtigung von Aufgabengebieten (§ 6 Abs. 4 HSchG):**

Rechtserziehung

## 1.2.3 Q3

Q3

**Konzepte und Anwendungen der Theoretischen Informatik**Std.:  
GK 36  
LK 63**Begründung:**

Die Schülerinnen und Schüler haben in der Einführungsphase (E1 und E2) sowie im 1. Jahr der Qualifikationsphase (Q1 und Q2) in Ansätzen Elemente der theoretischen Informatik kennen gelernt. Die vertiefte Auseinandersetzung mit relevanten theoretischen Konzepten im Kurshalbjahr Q3 baut auf diesen Vorkenntnissen auf und stellt Bezüge zu den im Informatikunterricht behandelten Themen her. Die Theorie ordnet Erfahrungen durch Systematisierung, weist Zusammenhänge auf und begründet bzw. beweist sie. Sie schafft durch ihre Begrifflichkeit die Voraussetzungen, um über die prinzipiellen Grenzen von Technik und menschlichem Geist etwas aussagen zu können. Die Ergebnisse der Theorie sind von beständigem Wert und nicht den schnellen Entwicklungszyklen und kurzen Halbwertszeiten der hektischen Computerwelt unterworfen. Die Inhalte der theoretischen Informatik sind nicht immer so direkt anwendbar wie viele Inhalte anderer informatischer Themenbereiche. Dafür sind die Erkenntnisse der theoretischen Informatik oft allgemeiner, umfassender und weitreichender als in anderen Gebieten der Informatik. Gemäß dem Grundsatz „Nichts ist praktischer als eine gute Theorie“ gilt es, die Bedeutung der Theorie durch deren Anwendung in der Praxis auf Schulniveau zu verdeutlichen. Die Behandlung theoretischer Fragestellungen darf nicht isoliert von den Anwendungen und auf Vorrat erfolgen. Übertriebene Formalisierungen gilt es zu vermeiden. Die Inhalte sind didaktisch reduziert darzubieten, ohne dabei die wesentlichen Einsichten zu verfälschen.

Die Informatik setzt sich mit informationsverarbeitenden Maschinen auseinander. Computersysteme unterstützen und erleichtern die geistige Arbeit von Menschen. Daher ist ein fundiertes Grundwissen über die Möglichkeiten und Grenzen der Organisation von Arbeit durch den IT-Einsatz zu vermitteln. Die Technische Informatik geht der Frage nach, wie informationsverarbeitende Maschinen gebaut werden können und wie die Hardware solcher Maschinen aussieht. Entsprechende Unterrichtsinhalte können im fakultativen Bereich behandelt werden. Die Theoretische Informatik setzt sich mit der Leistungsfähigkeit, den prinzipiellen Möglichkeiten und den Grenzen informationsverarbeitender Maschinen sowie den Grenzen der Problemlösung mit Algorithmen auseinander.

Die Theorie der Automaten und formalen Sprachen spielt eine wichtige Rolle bei der Entwicklung von Sprachen, z. B. Auszeichnungs-, Programmier- oder Datenbanksprachen, sowie von entsprechenden Übersetzer- und Interpreterprogrammen. Schülerinnen und Schüler haben in ihrer bisherigen Auseinandersetzung mit der Informatik mit diesen Programmen schon gearbeitet; nun findet eine Untersuchung ihrer Wirkprinzipien statt. Formale Sprachen sind Mittel der Kommunikation zwischen Mensch und Maschine. Der Zusammenhang zwischen formaler Sprache und erkennendem Maschinenmodell führt zur Chomskyschen Sprachhierarchie und damit zur Charakterisierung der formalen Sprachen durch Typen von Grammatiken.

Endliche Automaten als abstrakte Modelle realer Computer bilden einen möglichen Zugang zu theoretischen Grundlagen der Informatik. An ihnen lässt sich sowohl der Zusammenhang mit den von ihnen akzeptierten Sprachen als auch die prinzipielle Begrenztheit ihrer Leistungsfähigkeit zeigen. Kellerautomaten und Turingmaschinen stellen Automaten höherer Leistungsfähigkeit dar. Die Turingmaschine als universelle symbolverarbeitende Maschine repräsentiert nicht nur eine Präzisierung des Algorithmenbegriffs, sondern dient auch zur Veranschaulichung der prinzipiellen Grenzen maschineller Informationsverarbeitung.

Die Berechenbarkeitstheorie beschäftigt sich mit den grundsätzlichen Möglichkeiten und Grenzen der Algorithmisierbarkeit. Historisch gesehen steht sie am Anfang der Informatik, noch vor der Computer-Ära. Haupteinsicht dieser Theorie ist die Churchsche These, wonach die Klasse der im intuitiven Sinne berechenbaren Funktionen mit der Klasse der turing-berechenbaren Funktionen übereinstimmt und es nichtberechenbare Funktionen bzw. algorithmisch unlösbare Probleme gibt.

Im Grundkurs sollen die Fachbegriffe und Zusammenhänge möglichst anschaulich eingeführt werden, vertiefte mathematische Formalismen und Methoden sind zu vermeiden. Soweit möglich sind die theoretischen Konzepte mit konkreten Anwendungen zu verbinden.

Im Leistungskurs sind im Sinne wissenschaftspropädeutischen Arbeitens formalisierte und systematische Zugänge zu den Fachbegriffen und Zusammenhängen erforderlich; aber auch hier sollten zu weit gehende mathematische Formalismen und Methoden vermieden werden.

Die Modellierung einfacher realer Automaten (z. B. Getränkeautomat, Fahrkartenautomat, Aufzugmodell) führt zu den endlichen Automaten, die sich grafisch als Zustandsdiagramme darstellen lassen. Pfade auf dem Zustandsdiagramm werden durch die vom Automaten akzeptierten Wörter beschrieben. Die reguläre Grammatik ergibt sich aus der Umsetzung einzelner Zustandsübergänge in Produktionen. Zu den Automaten können die Schülerinnen und Schüler eigene Simulationsprogramme schreiben oder sie mit verfügbaren Programmen auf der Basis der Grammatiken simulieren.

Einfache Beispiele wie z. B. Scanner für Bezeichner, Zahlen oder Kontrollstrukturen führen schnell zu kontextfreien Grammatiken für reguläre Sprachen. Syntaxdiagramme schließen an die Vorkenntnisse der Schülerinnen und Schüler an und können als anschauliche Darstellungsmittel für Grammatiken genutzt werden. Anwendungsbeispiele stellen den Lebensweltbezug her. So können beispielsweise Elemente der in E1 und Q2 behandelten Sprachen HTML und SQL in Syntaxdiagrammen oder als Grammatiken dargestellt werden. Die in RFCs (Request for Comment) festgelegten Internetstandards bieten weitere Anwendungsbeispiele (z. B. RFC 821 – Simple Mail Transfer Protocol, RFC 1945 – Hypertext Transfer Protocol). Der Bereich der regulären Sprachen wird in den Beispielen im Allgemeinen erst dann verlassen, wenn Klammerterme bzw. rekursive Syntaxdiagramme auftreten.

Im Kurshalbjahr Q1 haben die Schülerinnen und Schüler schon die polynomiale und exponentielle Zeitkomplexität kennen gelernt. Beim Thema Berechenbarkeit geht es daher um den Nachweis, dass es nicht berechenbare Probleme gibt, und um die Unentscheidbarkeit des Halteproblems. Weitere Probleme, die auf das Halteproblem rückführbar sind, können behandelt werden.

Ein möglicher Zugang zum Thema Berechenbarkeit ist auf dem Hessischen Bildungsserver dokumentiert. Der Vergleich von Algorithmen mit polynomialer bzw. exponentieller Laufzeit führt schnell auf die Begriffe berechenbar und durchführbar. Erheblich mehr Aufwand ist für den Nachweis der Existenz nicht berechenbarer Funktionen nötig; schließlich muss man das Cantorsche Diagonalisierungsverfahren bemühen. Dagegen ist der Nachweis der Unentscheidbarkeit des Halteproblems mit deutlich weniger mathematischem Aufwand möglich.

Für den Leistungskurs und die fakultativen Inhalte sind noch folgende Aspekte bedeutsam:

Zur Vernetzung des Wissens sind die Bezüge zum Kurshalbjahr Q1 wichtig. Der abstrakte Datentyp Keller und die Methode der Rekursion können zur Simulation von Kellerautomaten benutzt werden. Erfahrungsgemäß gestaltet sich die Konstruktion eines Kellerautomaten zu einer kontextfreien Sprache schwierig, weswegen in der Literatur oft nur einfache Beispiele zu kontextfreien Sprachen vorkommen. Ein geeignetes Mittel ist beim Einsatz von Syntaxdiagrammen die Ersetzung rekursiver Aufrufe durch push-/pop-Operationen.

Die Turingmaschine hat eine einfachere Struktur als eine Registermaschine. Letztere sind aber leichter mit realen Mikroprozessoren vergleichbar. Exemplarisch kann man zeigen, dass mit Turingmaschinen arithmetische und logische Ausdrücke sowie alle Kontrollstrukturen verarbeitet werden können und sie somit genauso berechnungsstark wie Registermaschinen und moderne Computer sind.

Für die aktive Auseinandersetzung mit den theoretischen Maschinenmodellen sind Simulationen nützlich. Eine einheitliche Simulationsumgebung für Automaten, Kellerautomaten und Turingmaschinen steht zur Verfügung. Simulationsprogramme für Registermaschinen sind auch im Internet verfügbar.

Themen der theoretischen Informatik können gut mit der Programmiersprache Prolog bearbeitet werden. Formale Sprachen lassen sich einfach analysieren und generieren; Automaten können ohne großen Auf-

wand simuliert werden. Besonders geeignet ist die Verwendung von Prolog für den Übersetzerbau. Auf dem Hessischen Bildungsserver stehen Materialien zur Verfügung, die diesbezügliche schülergemäße Zugänge zeigen.

---

**Verbindliche Unterrichtsinhalte/Aufgaben:**

Formale Sprachen und Grammatiken	reguläre und kontextfreie Grammatiken und Sprachen Anwendung mit Syntaxdiagrammen Chomsky-Hierarchie (LK) kontextsensitive Sprachen (LK)
Endliche Automaten	Zustand, Zustandsübergang, Zustandsdiagramm Zeichen, Akzeptor Simulation realer Automaten (z. B. Getränkeautomat) Anwendung endlicher Automaten (z. B. Scanner) deterministische und nicht-deterministische Automaten (LK) reguläre Ausdrücke (LK) Mensch-Maschine-Kommunikation (LK)
Kellerautomaten (LK, GK fakultativ)	Automat mit Kellerspeicher kontextfreie Grammatiken Klammerausdrücke, Rekursion
Turing- oder Registermaschine (LK, GK fakultativ)	Turing- oder registerberechenbar Churchsche These Computer als universelle symbolverarbeitende Maschine Verhältnis Mensch – Maschine
Berechenbarkeit	Entscheidbarkeit, Halteproblem prinzipielle Grenzen algorithmischer Verfahren

---

**Fakultative Unterrichtsinhalte/Aufgaben:**

Übersetzerbau	Scanner, Parser, Interpreter und Compiler z. B. Steuersprache für Roboter, LOGO, Plotter oder miniPASCAL
Technische Informatik	Rechnerarchitektur, Assemblersprache (vgl. Wahlthema in Q4)
Komplexitätstheorie	die Klassen P und NP, P/NP-Problematik NP-vollständige Probleme

---

**Arbeitsmethoden der Schülerinnen und Schüler/Hinweise und Erläuterungen:**

Die Inhalte des Kurshalbjahres sollen in einem Projekt angewendet werden. Möglich sind z. B. Steuersprachen für Roboter oder Plotter, Interpreter für Formeln oder miniLOGO, Compiler für miniPASCAL oder mini-Java. Ein weiteres interessantes Anwendungsgebiet stellen Lindenmeyersysteme dar, mit denen biologisches Wachstum durch Grammatiken beschrieben wird. Möglich sind auch Projekte zur automatischen Sprachverarbeitung.

Im Leistungskurs soll ein Projekt vorzugsweise zum Thema Übersetzerbau stattfinden.

**Querverweise:****Formale Sprachen:** D**Datenbanken:** PoWi, G, Ek, Ch, M**Berücksichtigung von Aufgabengebieten  
(§ 6 Abs. 4 HSchG):**

**1.2.4 Q4**

Für den Kurs in Q4 ist ein Wahlthema vorgesehen. Angesichts der besonderen Situation im Prüfungshalbjahr ist es nicht notwendig, mit dem Wahlthema ein völlig neues Sachgebiet zu beginnen. Sinnvoll ist ein Wahlthema, das an die in den Halbjahren E1 bis Q3 gelegte Basis anschließt. Die zur Verfügung stehenden Wahlthemen reichen vom eher hardware-orientierten Ansatz über die Untersuchung spezieller Grundlagen bis hin zu aktuellen Anwendungsgebieten. In jedem Wahlthema ergeben sich Verknüpfungen zu anderen Wissensgebieten, so dass hier geeigneter Raum zur Diskussion und Untersuchung philosophisch-ethischer und fächerverbindender Aspekte oder dem Aufzeigen gesellschaftspolitischer oder wissenschaftlich relevanter Zusammenhänge ist.

Zur Vorbereitung auf das Abitur sollen bisher erworbene Kenntnisse in neuen Zusammenhängen verwendet oder vertieft und Querverbindungen zwischen den behandelten Themen erarbeitet und genutzt werden. Auf eine Wiederholung und Anwendung bekannter Verfahren darf sich der Kurs nicht beschränken.

Im Folgenden werden mehrere Wahlthemen vorgestellt. Auf eine Unterscheidung von verbindlichen und fakultativen Inhalten wurde bewusst verzichtet, um individuelle Schwerpunktsetzungen und vertiefte Projektarbeit zu ermöglichen. Daher werden auch keine expliziten Angaben zu den Arbeitsmethoden der Schülerinnen und Schüler gemacht. Die Aufzählung der möglichen Themen erhebt keinen Anspruch auf Vollständigkeit. Damit bleibt der Lehrplan für neuere Entwicklungen in der Informatik offen.

<b>Q4</b>	<b>Wahlthema: Betriebssysteme</b>	<b>Std.: GK 24 LK 43</b>
-----------	-----------------------------------	----------------------------------

**Begründung:**

Das Betriebssystem ist wesentlicher Bestandteil der Systemumgebung, in der Anwenderprogramme entwickelt und eingesetzt werden. Die Kenntnis von Grundlagen, Prinzipien und Aufgaben eines Betriebssystems ist daher für ein fundiertes Verständnis eines Hard- und Softwaresystems erforderlich. Da aktuelle Betriebssysteme sehr komplex sind, sollen spezifische Inhalte didaktisch reduziert mit geeigneten Modellen behandelt werden. Konzepte und Strategien zur Prozess-, Speicher- und Dateiverwaltung können unabhängig von einer speziellen Realisierung thematisiert werden. An geeigneten Stellen sollten die Schülerinnen und Schüler die Möglichkeit haben, mit einem neuen Betriebssystem (z. B. Linux) Erfahrungen zu sammeln.

Die bei Betriebssystemen verwendeten Konzepte (z. B. Schichten) und Strategien führen zu Datenstrukturen und Algorithmen (z. B. prioritätsgesteuerte Warteschlange), die teilweise aus anderen Kurshalbjahren bekannt sind und im Sinne eines Spiralcurriculums hier wieder aufgegriffen und vertieft werden können.

**Unterrichtsinhalte/Aufgaben:**

Schichtenmodell	Benutzer und Kernmodus Prozessvergabe, Speicherverwaltung, Prozesskommunikation, Ein-Ausgabe-Verwaltung Benutzerprogramme, Operateur
Prozesse und Tasks	Sequentielle Prozessmodelle Prozesszustände und Prozesstabelle Prozesskommunikation, kritische Bereiche Scheduling, Multitasking
Speicherverwaltung	Swapping Virtueller Speicher und Paging, Begriff der MMU Seitenersetzungsalgorithmen
Ein- und Ausgabeverwaltung	Interrupts und -behandlung durch Hard- und Software Softwareschnittstellen Gerätetreiber
Deadlocks Interprozesskommunikation	Unter- und ununterbrechbare Betriebsmittel Deadlockerkennung und -behebung Semaphoren Philosophenproblem
Dateiverwaltung	Dateien Verzeichnisse unter DOS und UNIX Zuverlässigkeitsbetrachtungen
Verteilte Systeme	Multiprozessorsysteme Netzwerkbetriebssystem

**Querverweise:**

**Welt- und Menschenbilder:** G, PoWi, Ek, Rka, Rev, Eth, Phil, Bio, Phy, E, F, Spa, Mu, Ku, GrA (Thema 4)

**Deterministisches Chaos:** M, Phy

**Evolution:** Bio, Phy, Rka, Eth

**Computersimulationen:** Bio, Ch, D, M, Phy

**Messen – Steuern – Regeln:** Phy, Ch, Spo

**Computergrafik:** Ku, M

**Berücksichtigung von Aufgabengebieten (§ 6 Abs. 4 HSchG):**

Q4

Wahlthema: Rechnernetze

Std.:  
GK 24  
LK 43**Begründung:**

Lokale und globale Rechnernetze bilden die Grundlage der Informations- und Kommunikationssysteme am Übergang vom zweiten zum dritten Jahrtausend. Das ISO/OSI-Referenzmodell liefert für die Auseinandersetzung mit dem Thema Rechnernetze einen sinnvollen Rahmen. Mit ihm kann das Prinzip der Rechnerkommunikation gut beschrieben werden. Im Unterrichtsverlauf wird man Schichten des Referenzmodells untersuchen und einzelne Aspekte vertiefend behandeln.

TCP/IP als Grundlage des Internets ist den Schülerinnen und Schülern schon aus E1 bekannt, so dass hier genauer zwischen den einzelnen Ebenen unterschieden werden soll. Sofern noch nicht geschehen, können die Schülerinnen und Schüler eigene Internetanwendungen auf der Basis von Sockets erstellen.

Technische Aspekte spielen bei der Betrachtung von Netzwerkstrukturen eine Rolle. Da verschiedene Medien (wie Kupfer, Glasfaser und Funk) in jedem Netz kombiniert werden können, müssen Übergangsmöglichkeiten geschaffen werden. Mit Hilfe des ISO/OSI-Referenzmodells lassen sich sowohl die verschiedenen Probleme als auch deren Lösungen gut darstellen.

Angesichts der Gefährdungen durch das Internet sollten auch sicherheitsrelevante Aspekte betrachtet werden. An einem Firewall lassen sich Schutzmöglichkeiten für lokale Netze aufzeigen. Wurde ein SQL-Server beim Thema Datenbanken nicht behandelt, bietet sich hier eine gute Gelegenheit dazu.

**Unterrichtsinhalte/Aufgaben:**

Kommunikationssysteme	ISO/OSI-Referenzmodell
Prinzipien und Protokolle	CSMA/CD TCP und IP, UDP
Dienste und Anwendungen	Kommunikation über Sockets und Ports POP3, SMTP, HTTP, FTP
Netzwerke	Topologien: Stern, Bus, Ring Arten und Medien: Ethernet (802.3, 802.11), BNC-Kabel, RJ45 (Twisted-Pair)-Kabel, Lichtwellenleiter, Funk Komponenten: Hubs und Switches
Routing	Router, NAT
Sicherheit	Prinzip des Firewalls
Weitere Dienste und Anwendungen	News, Timeserver, Proxies
Server	SQL-Server

**Querverweise:**

**Welt- und Menschenbilder:** G, PoWi, EK, Rka, Rev, Eth, Phil, Bio, Phy, E, F, Spa, Mu, Ku, GrA (Thema 4)

**Berücksichtigung von Aufgabengebieten (§ 6 Abs. 4 HSchG):**

Q4

Wahlthema: Computergrafik

Std.:  
GK 24  
LK 43**Begründung:**

Am Anfang des Kurses wird sicher – je nach der gewählten Programmiersprache – eine Phase notwendig sein, in welcher die zur Verfügung stehenden Möglichkeiten im grafischen Bereich untersucht werden. Dieses Thema bietet dann reichhaltige Möglichkeiten, sich mit verschiedenen Teilgebieten der Informatik zu befassen. Grafische Objekte lassen sich als Primitive definieren, die objektorientiert realisiert werden. Beim Übergang zu dreidimensionalen Grafiken kommen Anwendungen der Matrizenrechnung zum Zuge. Der Begriff der Perspektive lässt unterschiedliche Methoden der Umsetzung zu. Anspruchsvolle Clipping-, Hidden-Surface- oder Komprimierungs-Algorithmen können behandelt werden.

Die Grundlagen und Anwendungen in diesem Gebiet der Informatik geben weiterhin die Gelegenheit für fächerverbindenden oder fachübergreifenden Unterricht. Mathematik oder Kunst bieten sich an.

**Unterrichtsinhalte/Aufgaben:**

Computergrafik in 2 Dimensionen	Zeichenwerkzeuge und -möglichkeiten Grafische Primitive (Kreis, Quadrat, ...) Repräsentation von Objekten Bewegung Clipping
3D-Modelle	Transformation 3D auf 2D Perspektive
Optimierungsstrategien	Hidden Surface
Farbmodell	RGB, CMYK 24-Bit-Farbraum Paletten
Komprimierung	JPEG, GIF (LZW-Komprimierung) oder TIFF (LZH-Komprimierung)
Lichtquellen	Raytracing Radiosity
Softwareschnittstellen	Fluchtpunkte
Softwareschnittstellen	DirectX OpenGL

**Querverweise:**

**Welt- und Menschenbilder:** G, PoWi, Ek, Rka, Rev, Eth, Phil, Bio, Phy, E, F, Spa, Mu, Ku, GrA (Thema 4)

**Computergraphik:** Ku, M

**Berücksichtigung von Aufgabengebieten (§ 6 Abs. 4 HSchG):**

Q4

Wahlthema: Prolog als Sprache der künstlichen Intelligenz

Std.:  
GK 24  
LK 43**Begründung:**

In diesem Kurs kann – sofern noch nicht an einer anderen Stelle geschehen – die Behandlung einer zweiten Programmiersprache mit einem nicht imperativen Sprachparadigma erfolgen. Prolog bietet sich als deklarative Sprache aus mehreren Gründen an:

- sie ist syntaktisch einfach, beinhaltet aber die Mechanismen der Unifikation, Resolution und des Backtrackings,
- die Datenstruktur *Baum* ist in Prolog implementiert,
- gerade in den Gebieten *künstliche Intelligenz* sowie *Expertensysteme* zeigt sich, dass eher die logischen Aspekte des Problems als die Implementierung einer Lösung im Vordergrund der Arbeit stehen.

Die Arbeitsumgebung (Interpreter etc.) kann kostenlos über das Angebot des Hessischen Bildungsservers bezogen werden; weitergehende Materialien (Provisor etc.) unterstützen den Lernprozess.

**Unterrichtsinhalte/Aufgaben:**

Grundlagen	Fakten, Regeln, Prädikate, rekursive Prädikate
Ablaufverfolgung	Trace, Vierport-Modell
Datenbanken und einfache Anwendungen	Listen, Arithmetik, Ein- und Ausgabe
Backtracking	Fail und cut
Unifikation	Unifikationsdefinition und -regeln
Expertensysteme	Wissensbasis, Regelsysteme
Suchverfahren	Breitensuche, Tiefensuche
Modellierung mit Prolog	Grammatiken, formale Sprachen, Automaten

**Querverweise:**

**Welt- und Menschenbilder:** G, PoWi, Ek, Rka, Rev, Eth, Phil, Bio, Phy, E, F, Spa, Mu, Ku, GrA (Thema 4)

**Welt- und Menschenbilder:** G, PoWi, Ek, Rka, Rev, Eth, Phil, Bio, Phy, E

**Berücksichtigung von Aufgabengebieten (§ 6 Abs. 4 HSchG):**

Q4

Wahlthema: Simulationen, Chaostheorie

Std.:  
GK 24  
LK 43**Begründung:**

Ist eine Möglichkeit zum fächerverbindenden Arbeiten gegeben, kann das Thema Simulationen ins Auge gefasst werden. Simulationen nehmen einen breiten Raum in den Anwendungen der Informatik ein. Ihre Grundlagen bestehen im Allgemeinen sowohl aus einem mathematischen wie auch einem fachwissenschaftlichen Anteil. Das Räuber-Beute-Modell in der Biologie auf der einen Seite, die Entwicklung finanzmathematischer Zusammenhänge und die Betrachtung von Fraktalen am anderen Ende zeigen das Spektrum der Möglichkeiten, welche die Behandlung dieses Themas bietet.

Die informatischen Inhalte in diesem Gebiet können sich unter anderem aus der Algorithmik, der Optimierung des Laufzeitverhaltens und der grafischen Ausgabe rekrutieren. Auch die Umsetzung spezieller mathematischer Methoden ist ein lohnendes Gebiet, welches wieder in das Feld der Datenstrukturen und Objekte führen kann. Fächerverbindend mit beispielsweise Mathematik oder Biologie oder Wirtschaftswissenschaften ist folgende exemplarische Auswahl von Inhalten realisierbar:

**Unterrichtsinhalte/Aufgaben:**

Modellbildung	Realität → Modell → Simulation
Abgeschlossenes System	Räuber-Beute-Modell Differenzgleichungen Grafische Umsetzung
Rückkopplungskreise	Ursache-Wirkungs-Diagramme Eskalierender und stabilisierender Kreis Bevölkerungsmodell als Kombination Begrenztes Wachstum
Chaotische Systeme	Komplexe Zahlenfolgen/Iterationsverfahren Mandelbrot-Menge Fraktale Grafische Darstellung

**Querverweise:**

**Welt- und Menschenbilder:** G, PoWi, Ek, Rka, Rev, Eth, Phil, Bio, Phy, E, F, Spa, Mu, Ku, GrA (Thema 4)

**Evolution:** Bio, Phy, Rha, Eth

**Computersimulationen:** Bio, Ch

**Berücksichtigung von Aufgabengebieten (§ 6 Abs. 4 HSchG):**

Q4

Wahlthema: Technische Informatik

Std.:  
GK 24  
LK 43**Begründung:**

Grundlagen aus dem Themenbereich der Technischen Informatik leisten einen Beitrag für das Verständnis der Funktionsprinzipien von Informatiksystemen als symbolverarbeitende, universelle Maschinen. Im Vordergrund steht die exemplarische Auseinandersetzung mit den Ideen und den Prinzipien der Computersysteme, die sie als technisches Produkt ermöglichen. Dabei soll ausgehend von den historischen Entwicklungslinien und ihrer Realisierung auf unterschiedlichen technischen Ebenen abstrahiert werden. Chancen, Risiken und Folgen bei der Entwicklung zur Informationsgesellschaft und der sich damit verändernden Lebens- und Arbeitsformen lassen sich hierbei aufzeigen.

In der Regel wird man mit den logischen Grundsaltungen beginnen und mit Volladdieren in Form von Blockschaltbildern ein einfaches Addier- und Subtrahierwerk mit Komplementbildung realisieren. Die Schaltnetze können gut mit Simulations-Programmen aufgebaut und analysiert werden. Nach der Behandlung von einfachen Speicher-Bausteinen wird man das „Von Neumann Prinzip“ als Abstraktion und Grundlage moderner Mikroprozessoren einführen. Anhand des Blockschaltbildes und von Simulationen eines Mikroprozessors (z. B. 8086) werden dessen Grundprinzipien erarbeitet. Der Kurs soll mit der Analyse einiger beispielhafter Assemblerprogramme abgerundet werden.

**Unterrichtsinhalte/Aufgaben:**

Logische Grundsaltungen	AND, OR, NOT, NAND, NOR, EXOR, Arbeitstabelle, Symbol
Binärcodierung	Binärcode, BCD-Code, HexCode Subtraktion durch Komplementbildung
Einfache Rechenwerke	Halbaddierer, Volladdierer 4-Bit Addier- und Subtrahierer
Speicherelemente	RS-Flip-Flop, D-Flip-Flop, JK-Master-Slave-Flip-Flop Anwendungen, DEA
Von-Neumann-Prinzip	Funktionseinheiten, Speichermodell Befehlsabarbeitung
Mikroprozessor- und Mikrocomputerarchitektur	Bussysteme, ALU, Register, Flag-Register Steuerwerk, Mikroprogramm BIOS, ROM, RAM
Assemblerprogrammierung	Entwicklungswerkzeuge, Beispiele Adressierungsarten Codeschablonen für Schleifen und Fallunterscheidungen Interrupts
Technische Realisierung von Grundsaltungen	Relais-, Transistorschaltungen
Vereinfachung von Schaltnetzen mit boolescher Algebra	Gesetze der Booleschen Algebra konjunktive und disjunktive Normalform

**Querverweise:**

**Welt- und Menschenbilder:** G, PoWi,  
Ek, Rka, Rev, Eth, Phil, Bio, Phy, E, F,  
Spa, Mu, Ku, GrA (Thema 4)

**Messen:** Phy, Ch, M

**Berücksichtigung von Aufgabengebieten  
(§ 6 Abs. 4 HSchG):**

## 2 Abschlussprofil am Ende der Qualifikationsphase

Zentrales Ziel des Informatikunterrichts ist die Ausbildung der Sach-, Methoden-, Sozial- und Selbstkompetenz im Umgang mit Informations- und Kommunikationssystemen, welche die Schülerinnen und Schülern zur gleichberechtigten und aktiven Teilhabe am gesellschaftlichen Leben befähigen sowie auf Studium, Beruf und Gestaltung des individuellen Lebens in der Informationsgesellschaft vorbereiten.

### Allgemeine Ziele:

- Persönlichkeitsentwicklung des Einzelnen durch Förderung seiner Urteils- und Handlungsfähigkeit in der Informationsgesellschaft und die Entwicklung eines verantwortungsbewussten Umgangs mit Informatiksystemen.
- Vermittlung grundlegender Wirkprinzipien von Informatiksystemen und ihrer Beiträge zur Entwicklung von Kultur und Wissenschaft.
- Einordnung der Voraussetzungen, Chancen, Risiken und Folgen bei der Entwicklung zur Informationsgesellschaft und der damit sich verändernden

### Fachspezifische Ziele und Kenntnisse:

- Kenntnis und Anwendung von Konzepten, Aufbau und Wirkprinzipien wichtiger Informatiksysteme wie Internet, Programmierumgebung und Datenbanken
- Kenntnis und Berücksichtigung rechtlicher Rahmenbedingungen, insbesondere Urheber- und Datenschutzrecht, als Basis verantwortlichen Umgangs mit der Informations- und Kommunikationstechnik
- Daten- und Kontrollstrukturen sowie Objekte, Klassen und Klassenbeziehungen zum Problemlösen durch Programmierung
- Aufbau eines Repertoires wichtiger abstrakter Datentypen und iterativer und rekursiver Standardalgorithmen zum Bearbeiten von Datenstrukturen und Absuchen von Lösungsräumen
- Grundlagen relationaler Datenbanksysteme, Modellierung mit ER-Diagrammen, Umsetzung in das Relationenmodell, Abfragen mit QBE und SQL
- Theoretische Fundierung – der Computer als symbolverarbeitende Maschine, Grenzen des Computers: Grammatiken, formale Sprachen und Automaten, Berechenbarkeit
- Auseinandersetzung mit normativen, ethischen und sozialen Aspekten der IT-Technik und dem Verhältnis von Technik und Gesellschaft
- LK: Kenntnis einer prädikativen oder funktionalen Programmiersprache

### Methoden:

- Kenntnis und Anwendung von Verfahren der informatischen Modellierung bei Hypertextsystemen, Client-Server-Systemen, Programmierung und objektorientierter Problemlösung, Datenbanken, Automaten und formalen Sprachen
- Einsatz von UML-Diagrammen zur graphischen Darstellung von Modellierungen, systematische Umsetzung der Diagramme in programmiersprachliche Notation

- Strukturierte Dokumentation von Arbeitsergebnissen mit computerbasierten Medien, insbesondere als Hypertext
- Systematisches Problemlösen durch Analyse der Problemsituation, Modellierung eines Informatiksystems, Konstruktion, Test, Anwendung, Revision und ,Dokumentation einer Lösung
- Verständnis und Nutzung komplexer digitaler Dokumentationen (z. B. Hilfesysteme, Klassenhierarchien) zum selbstständigen Lernen und Problemlösen
- Erweiterung der Kooperations- und Kommunikationsfähigkeit durch projektorientiertes Arbeiten in Teams, sachgerechte mediale Aufbereitung der Ergebnisse in Schriftform und Vortrag im Plenum auf der Basis angewandter Präsentationstechnik
- Nutzung elektronischer Kommunikation (z. B. E-Mail, News, Groupware) im Lehr-Lernprozess

Der Grundkurs vermittelt grundlegende Kenntnisse der fachwissenschaftlichen Methoden und einen Überblick über ihre Anwendung auf wesentliche Bereiche innerhalb und außerhalb der Informatik.

Der Leistungskurs Informatik orientiert sich stärker an der Systematik der Fachwissenschaft und vermittelt ein vertieftes Verständnis und erweiterte Kenntnisse, hat insgesamt also eine wissenschaftspropädeutische Ausrichtung.